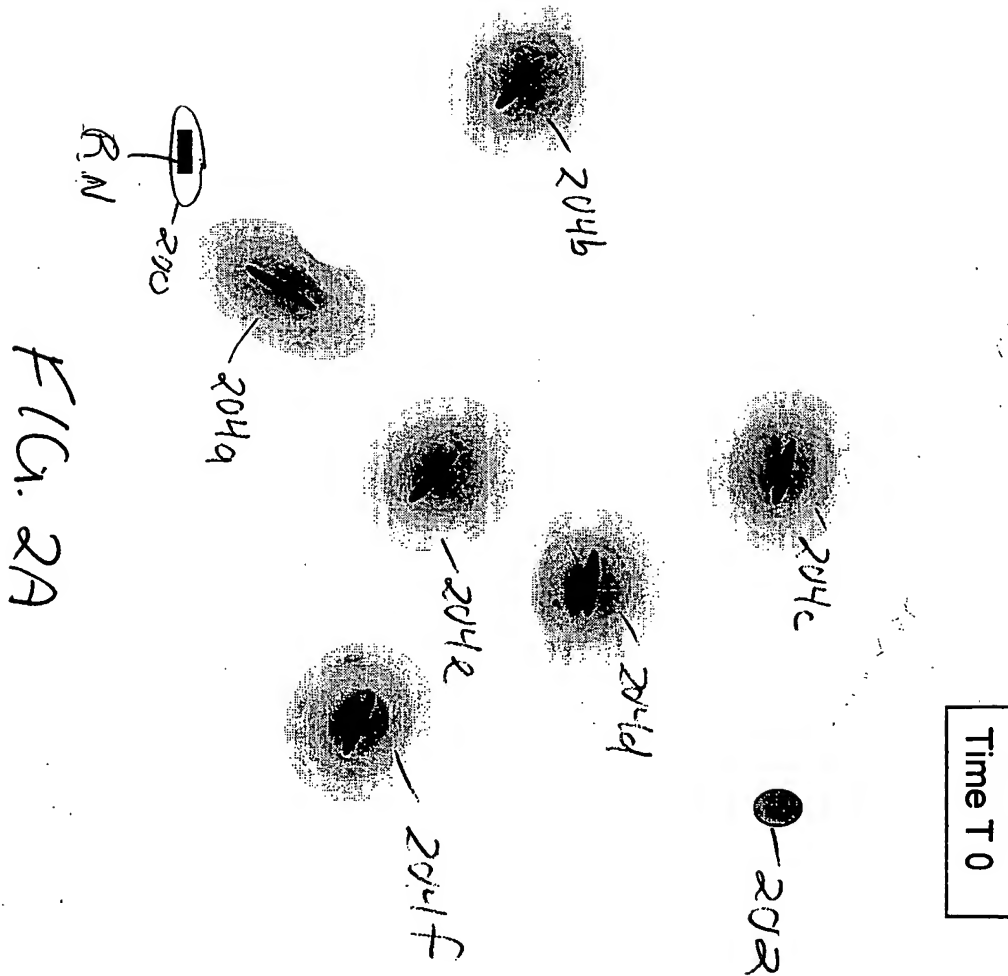


FIG. 1

SYSTEM AND METHOD FOR ADAPTIVE PATH PLANNING

Regina Estkowski, et al.
RTN-208PUS

2/21



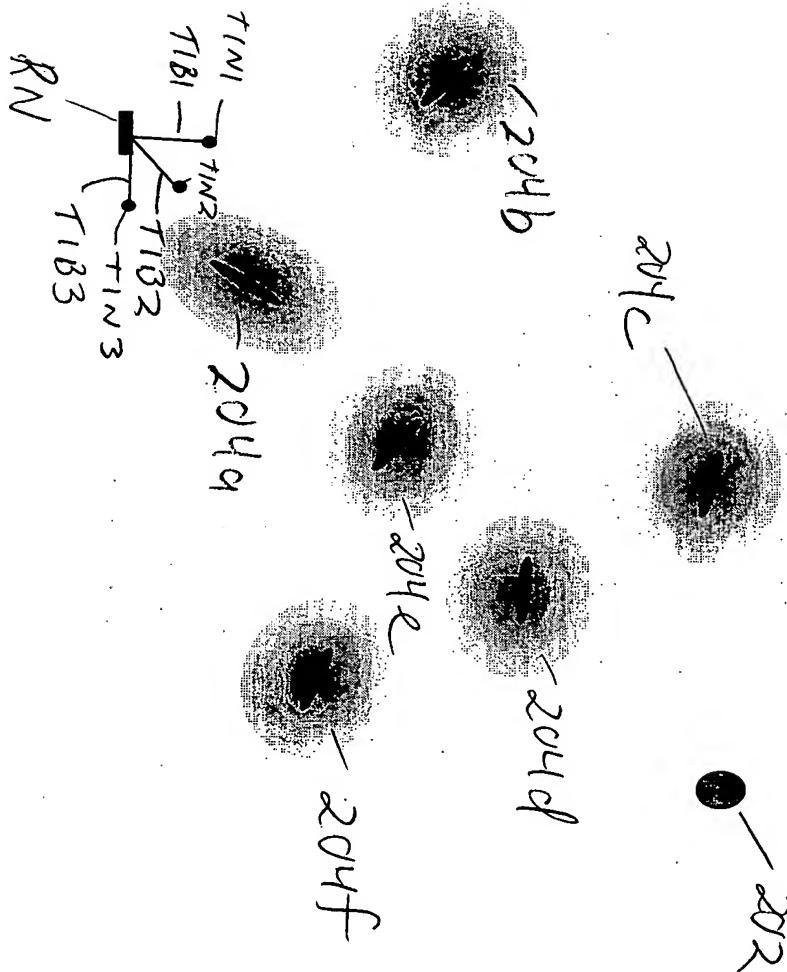
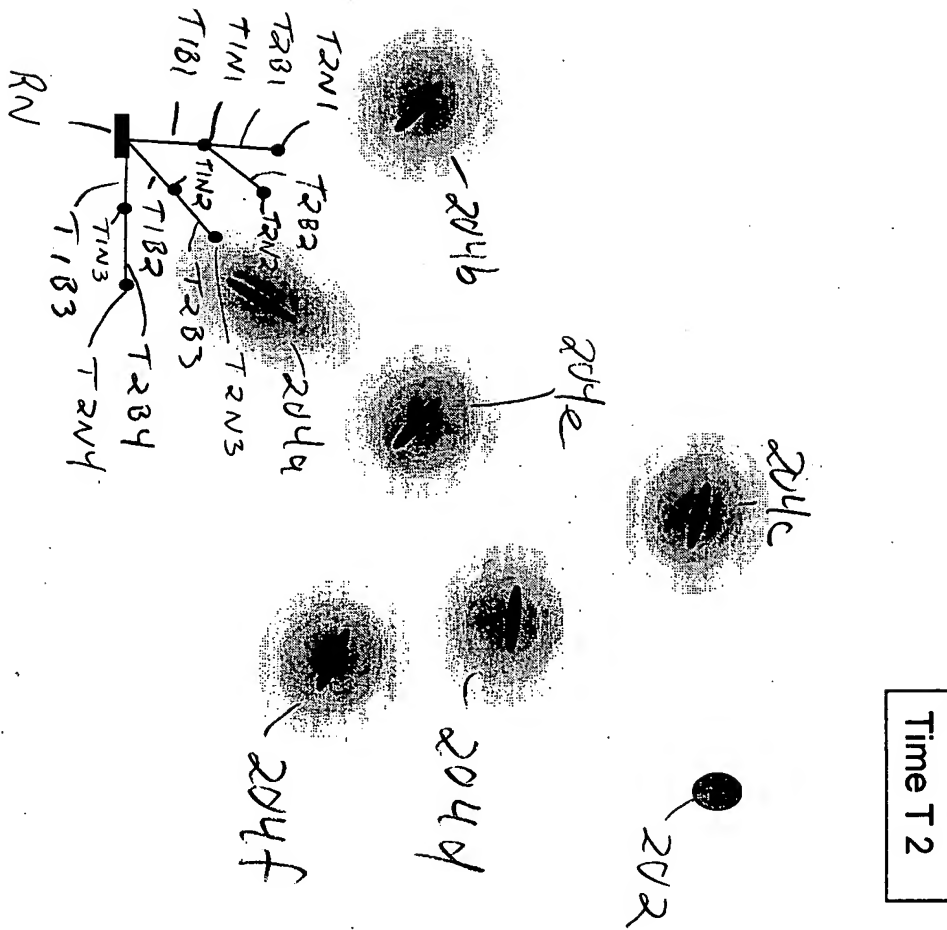


FIG. 23



5/21

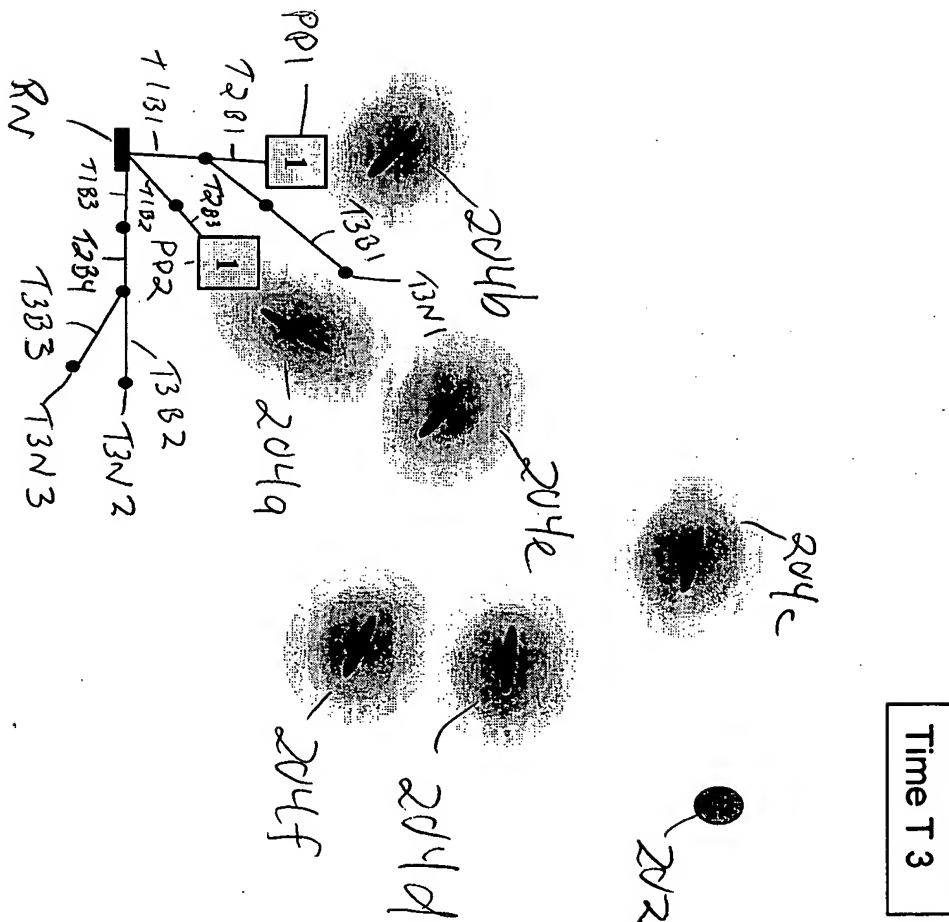


FIG. 20

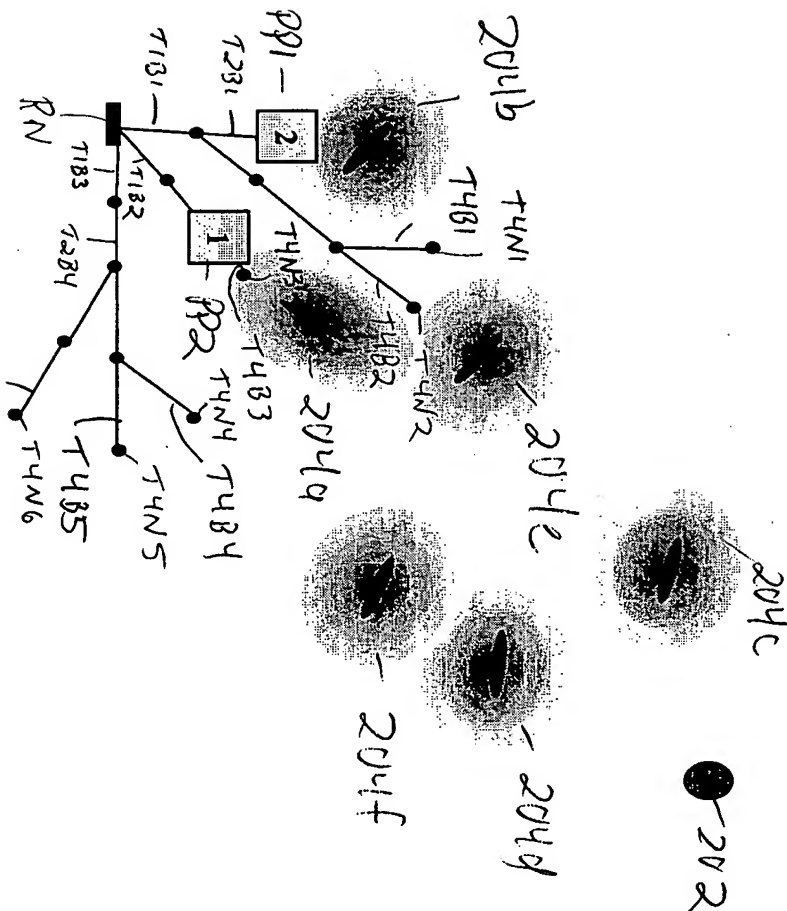


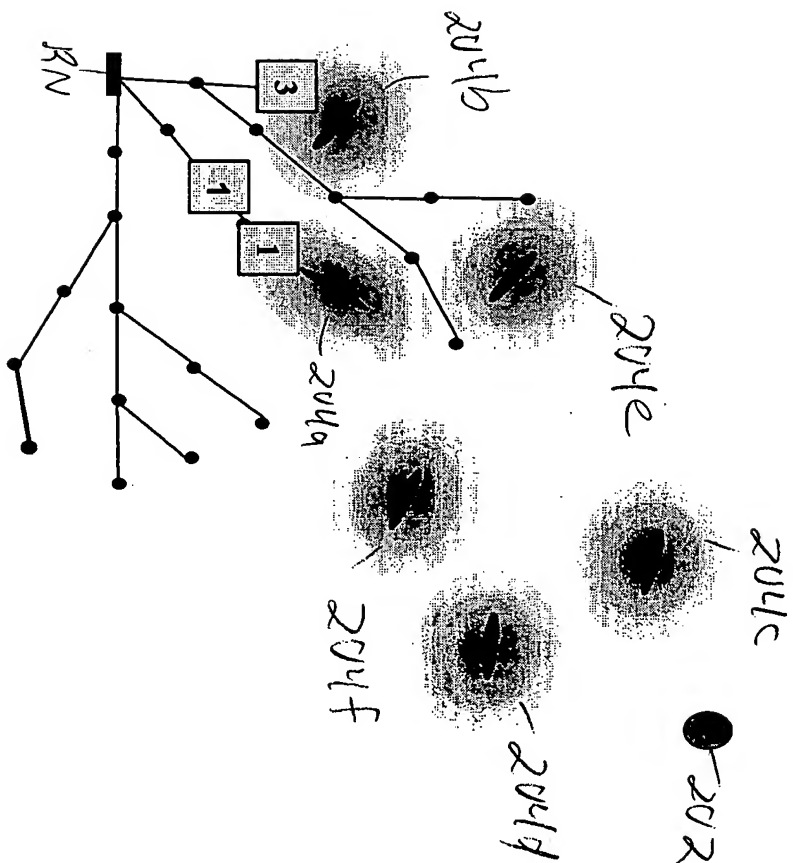
FIG. 2

SYSTEM AND METHOD FOR ADAPTIVE PATH PLANNING

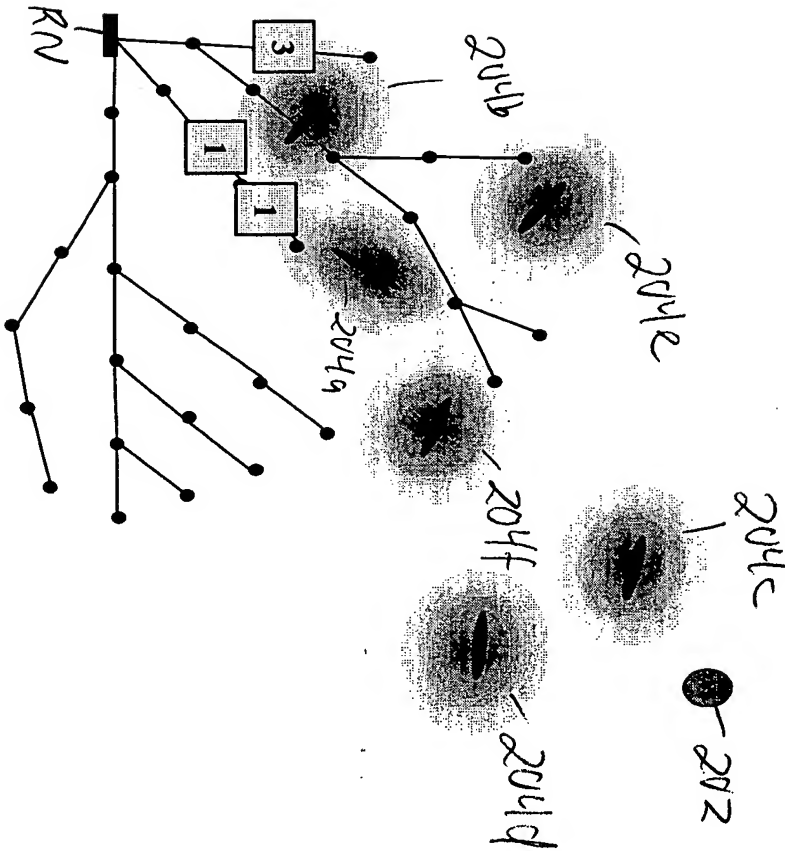
Regina Estkowski, et al.
RTN-208PUS

7/21

FIG. 2F

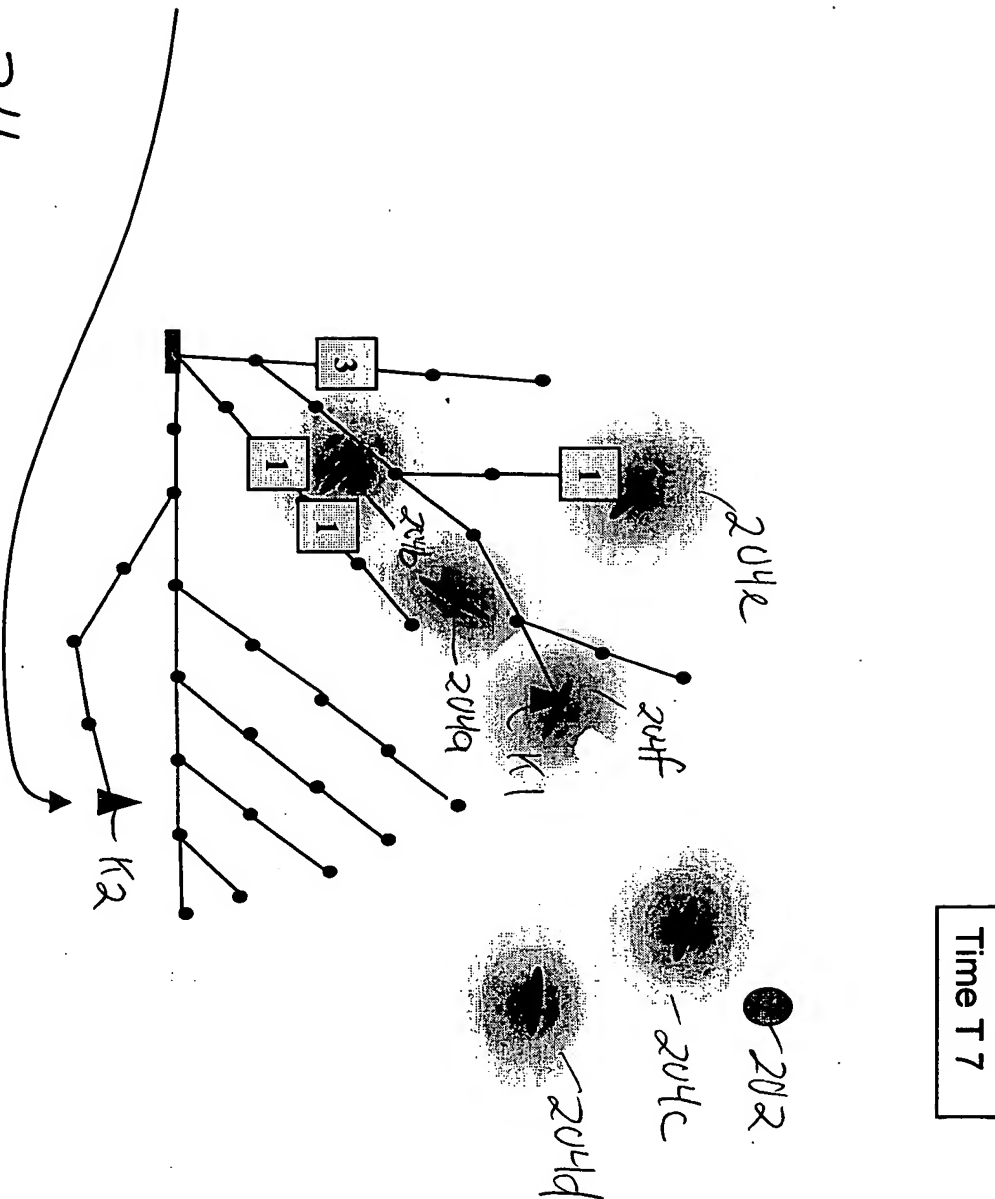


F1G. 2G



9/21

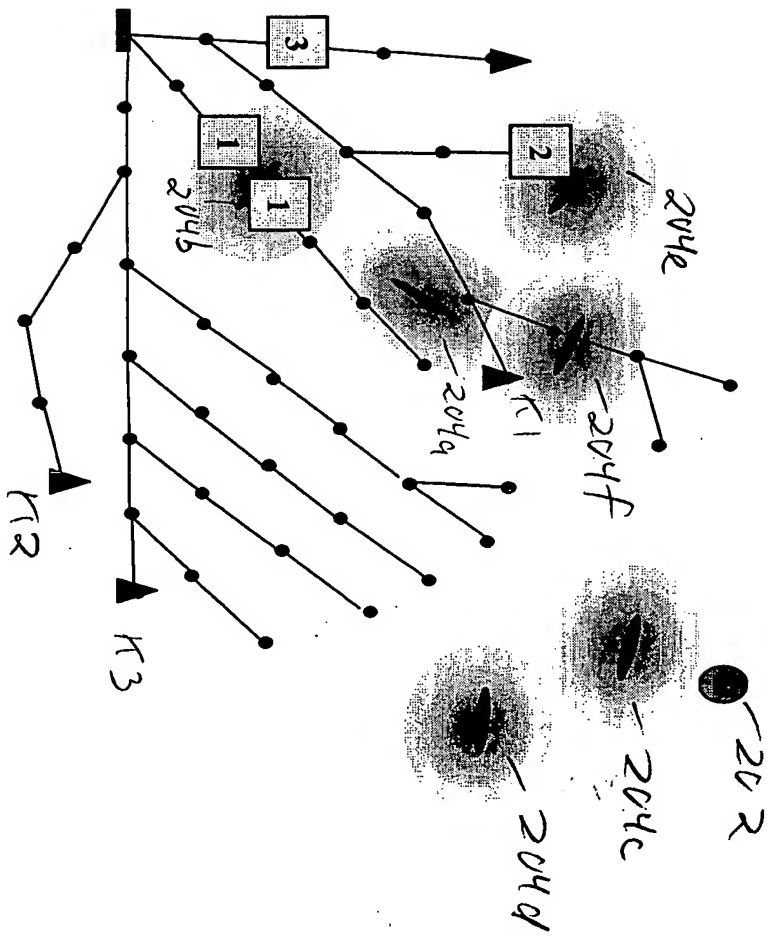
FIG. 2H



SYSTEM AND METHOD FOR ADAPTIVE PATH PLANNING

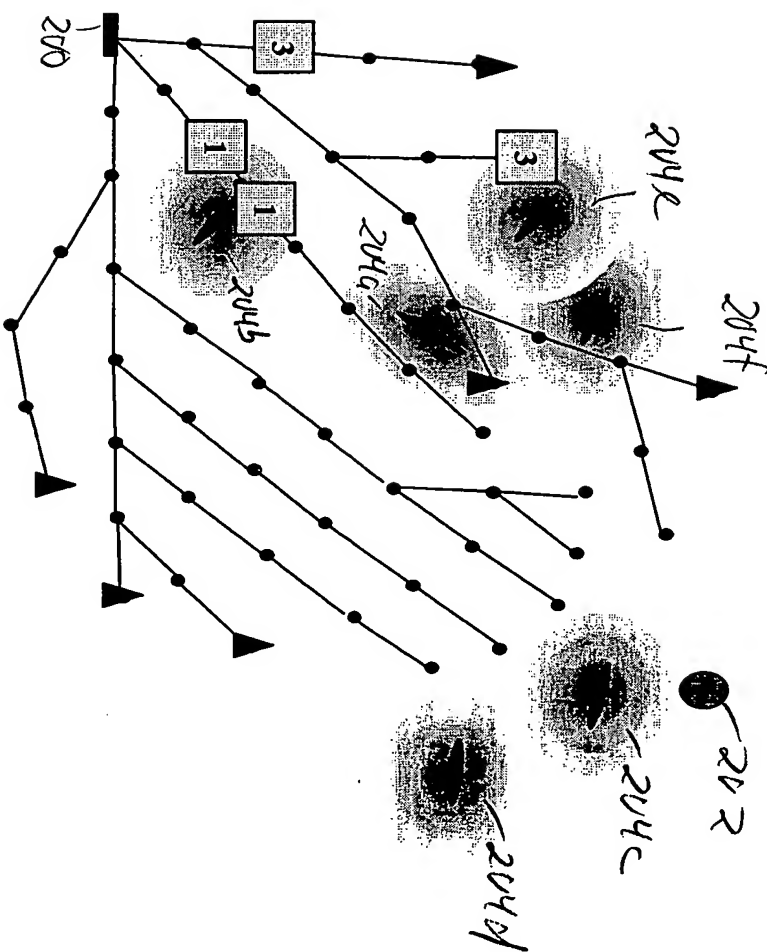
Regina Estkowski, et al.
RTN-208PUS

10/21



F-16. 21

FIG. 25



12/21

FIG. 2K

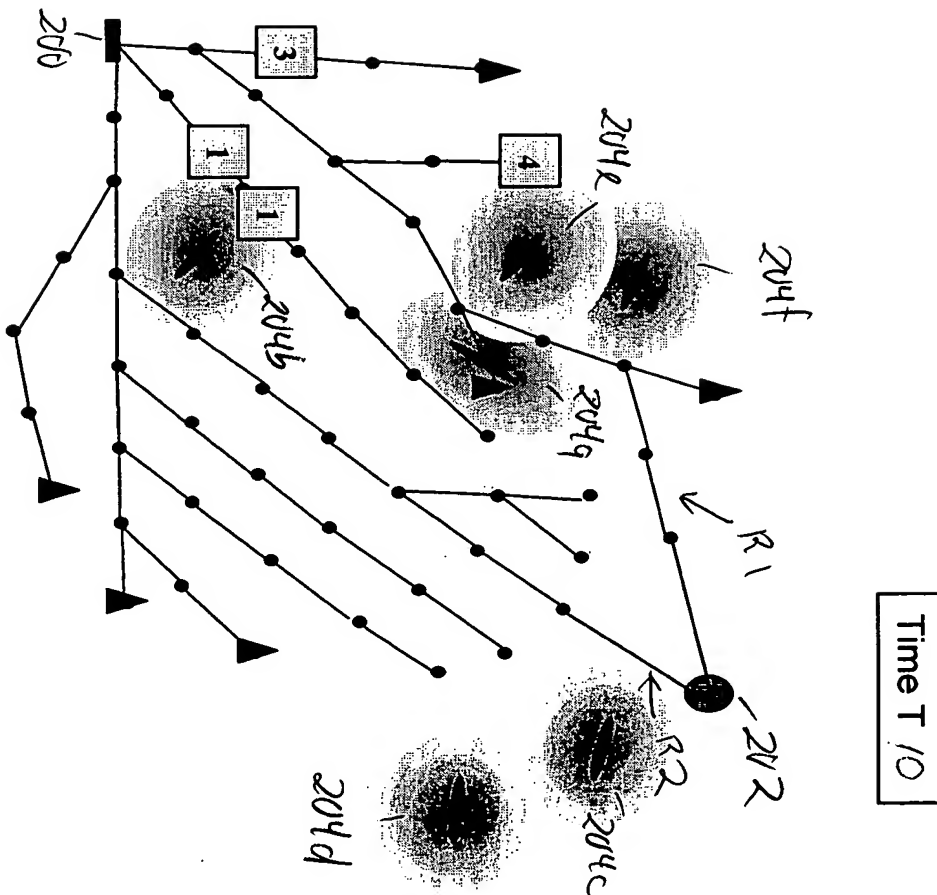
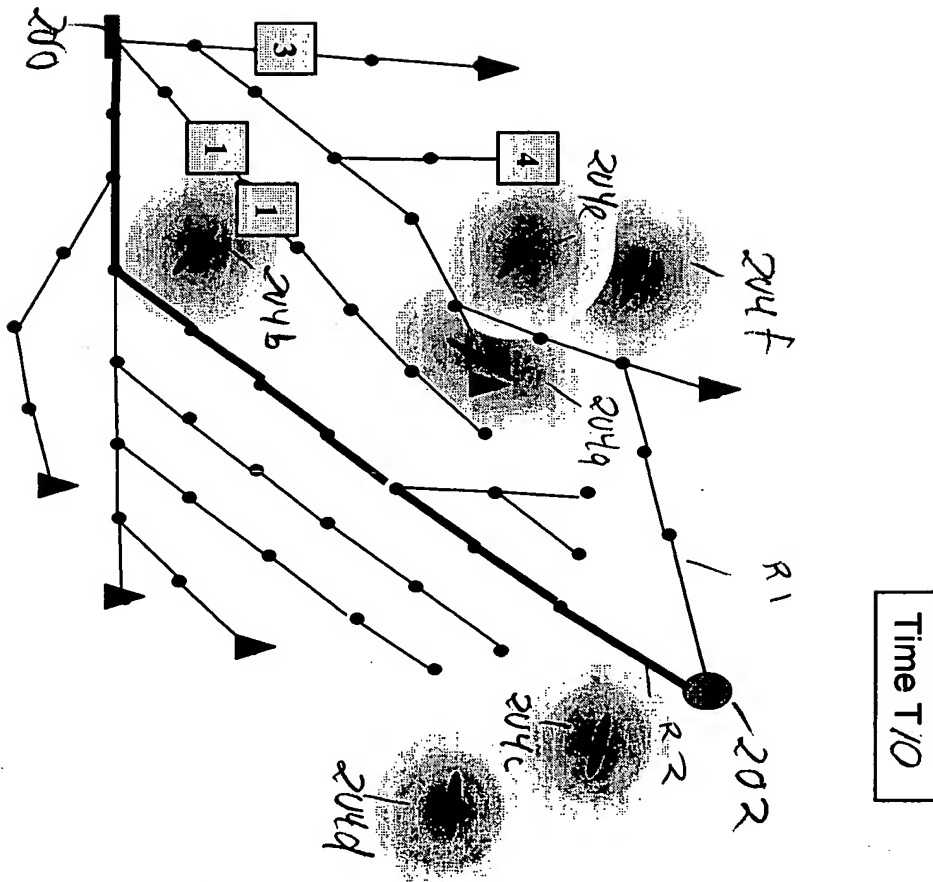


FIG. 2L



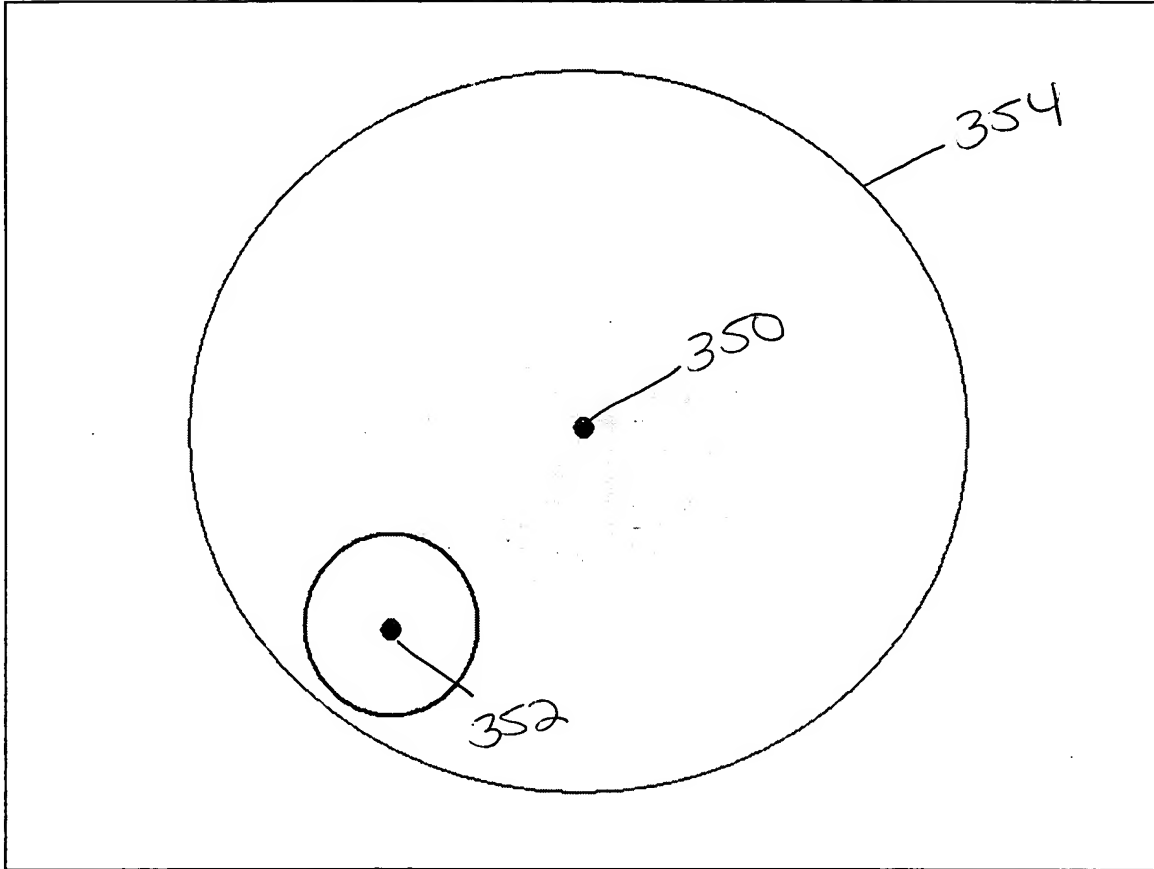


FIG 3 A

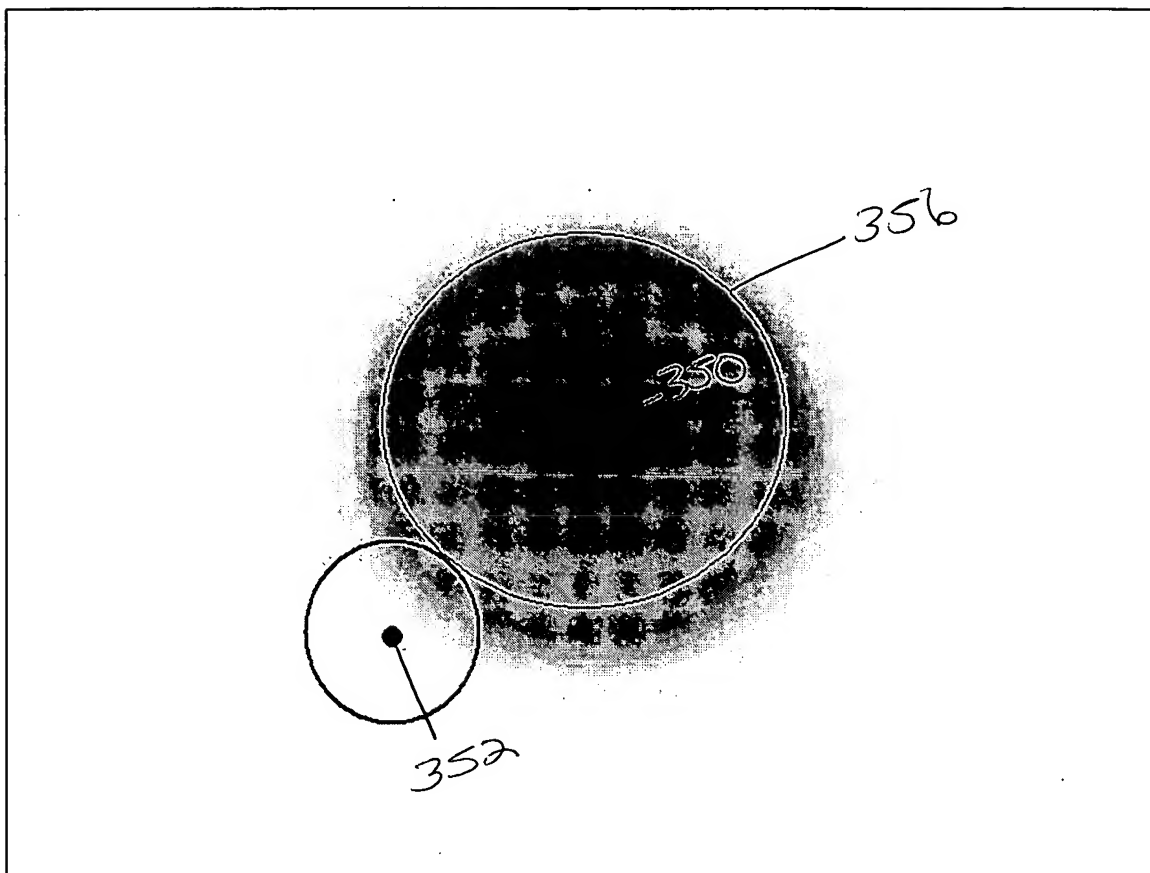


FIG 3B

16/21

400

WHILE the stopping conditions given by *Method Component 8* are not satisfied

 DO *Deterministic Tree Extension*.

 DO *Random Tree Extension*.

 Set all leaf nodes that have not been extended in 3. or 4. to DEAD.

END WHILE

Deterministic Tree Extension

FOR each leaf node, N , that is not DEAD

 Apply *Method Component 5* and obtain a set, X , of candidate path extensions to N .

 FOR each candidate path extension, $\pi \in X$

 Apply *Method Component 7* to determine if π is feasible.

 IF π is feasible THEN extend N by π .

 END FOR

END FOR

Random Tree Extension

WHILE *Method Component 5* says to continue random extension

 Apply *Method Component 5* to obtain a set, Λ , of candidate nodes for random extension.

 FOR each node, $N \in \Lambda$, apply *Method Components 5 and 6* to obtain a set, X , of candidate path extensions

 to N .

 FOR each candidate path extension, $\pi \in X$

 Apply *Method Component 7* to determine if π is feasible.

 IF π is feasible THEN extend N by π .

 END FOR

END WHILE

FIG. 4

SYSTEM AND METHOD FOR ADAPTIVE PATH PLANNING

Regina Estkowski, et al.
RTN-208PUS

17/21

300

1. Add the root node at your current position to *T*;
 2. Obtain the current *Turn Wedge* from the *VMM*;
 3. FOR each of *#SN* directions determined by discrete uniform distribution in the *Turn Wedge*, attempt to extend in the direction;
 4. END FOR;
 5. IF it is not possible to extend in all *#SN* directions
 6. THEN chose at most *#RA* random directions within the *Turn Wedge* and attempt to extend in these directions until *#SN* extensions have been attained;
 7. END IF;
 8. WHILE (*Stop Flag* == FALSE)
 - Set *Active Leaf List* = *New Leaf List*;
 - Set *New Leaf List* to Empty;
 - FOR each *Active* leaf node, *N*, in *T*
 - Attempt to extend straight ahead from *N*;
 - Attempt to extend towards the goal from *N*;
 - END FOR;
 - FOR each of the, at most, *#RN* *Active* leaf nodes having the best
- NM*
- Obtain the current *Turn Wedge* from the *VMM*;
 - Choose *#RE* random directions within the *Turn Wedge* and attempt to extend in each of these directions;
 - IF it is not possible to extend in all *#RE* directions
 - THEN chose at most *#RA* random directions within the *Turn Wedge* and attempt to extend in these directions until *#RE* extensions have been attained;
 - END IF;
 - END FOR;
 - IF the *Stop Condition* has been met
 - Set *Stop Flag* = TRUE;
- END WHILE;

FIG. 5

18/21

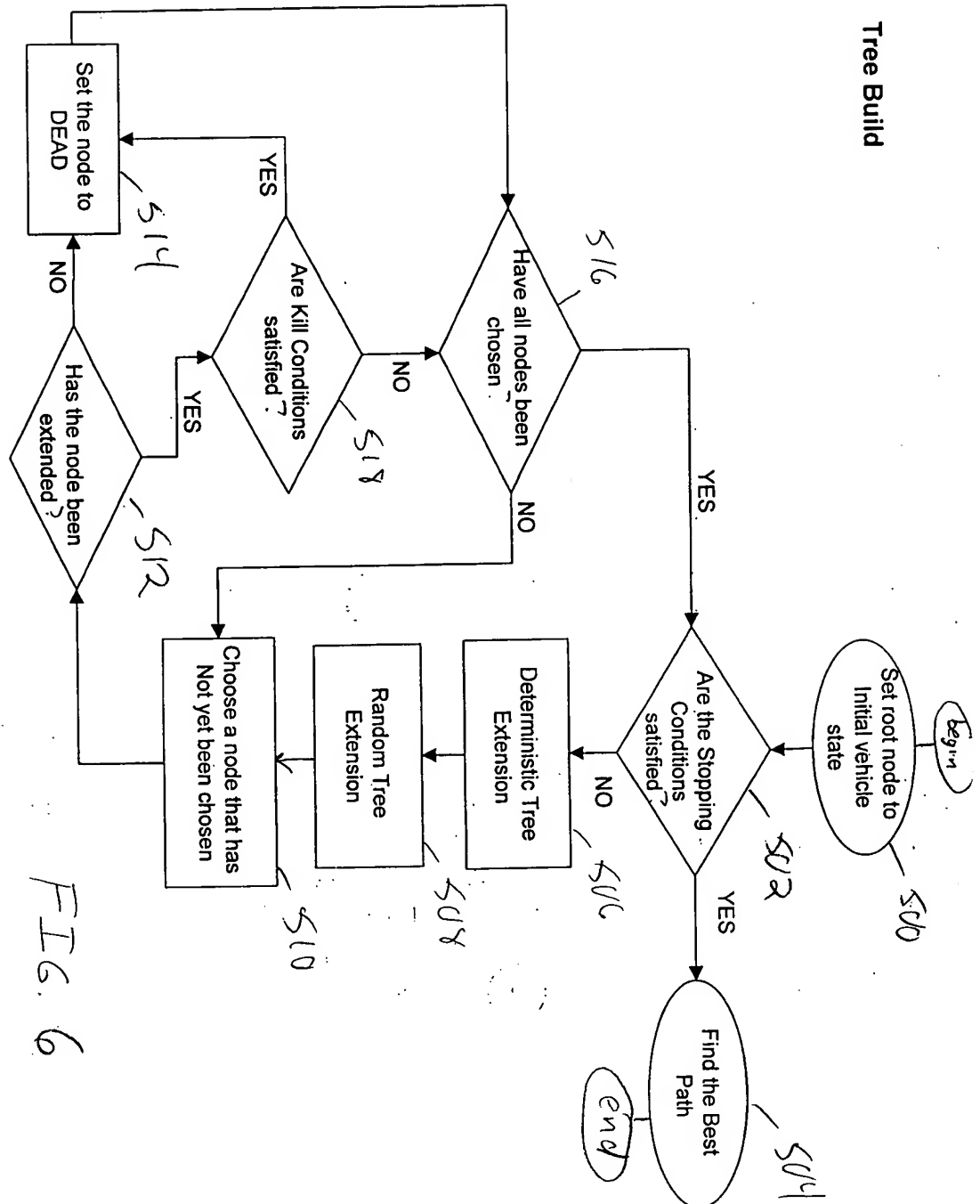


FIG. 6

**Deterministic Tree
Extension**

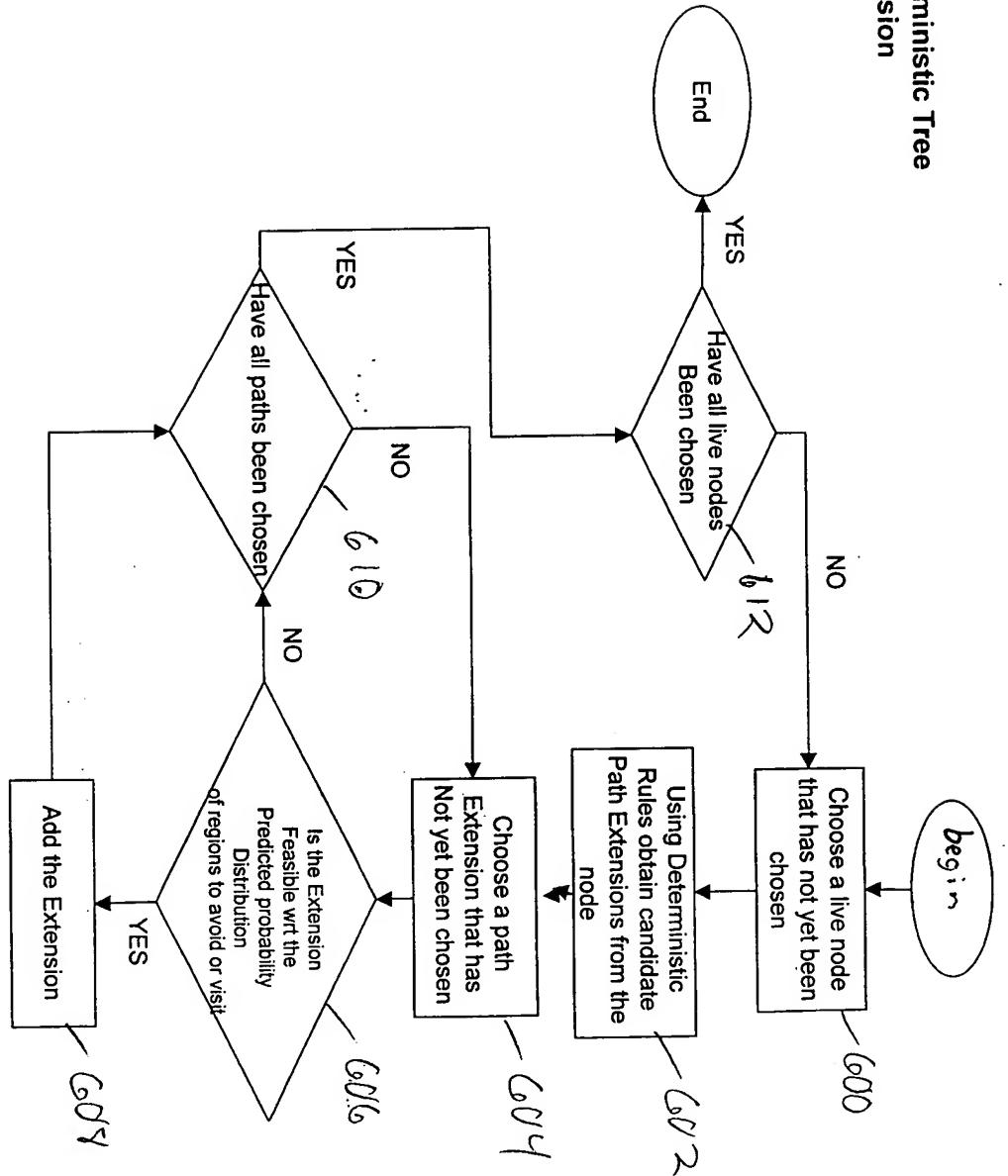


FIG. 7

*Random Tree
Extension*

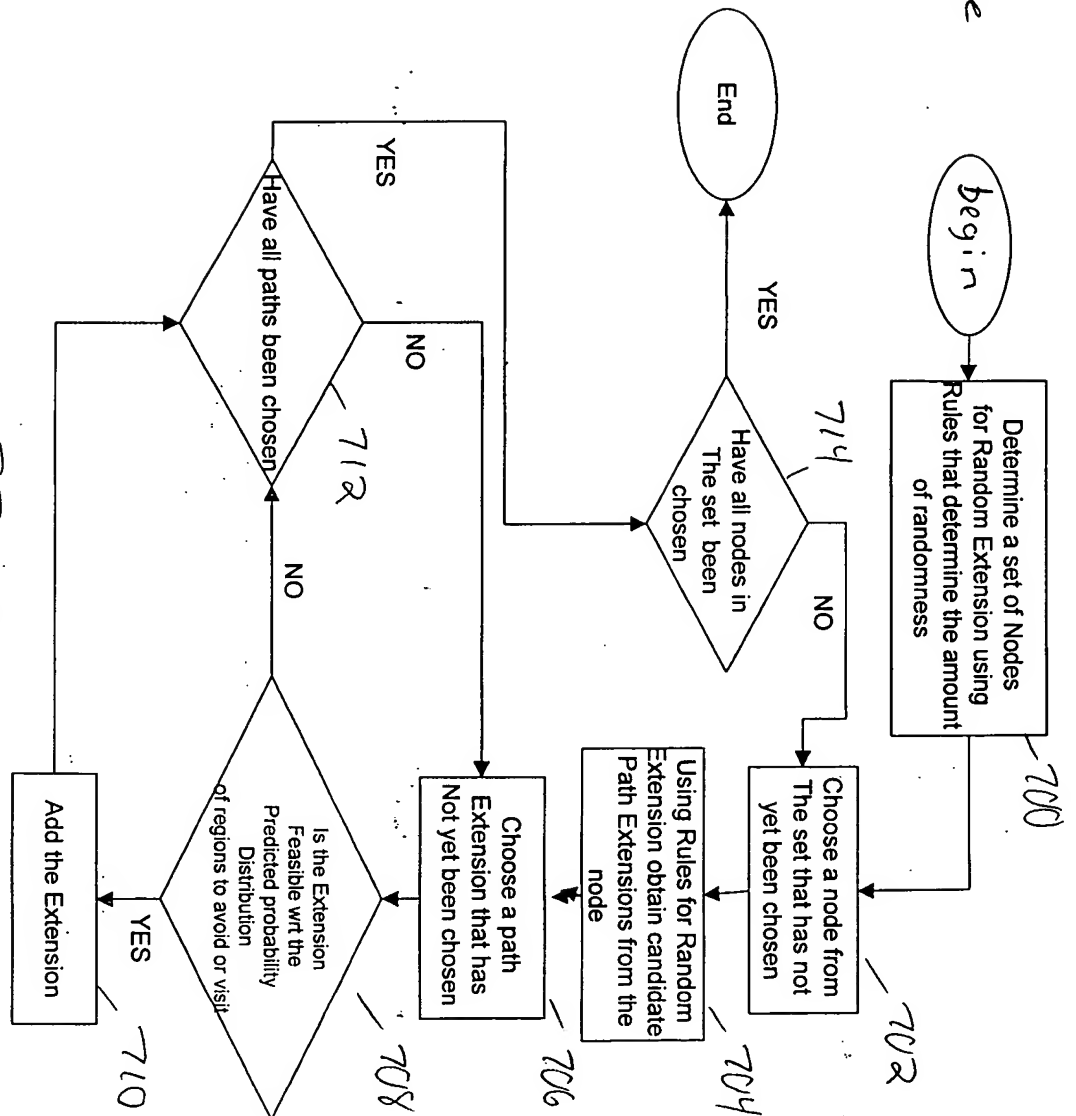


FIG. 8

SYSTEM AND METHOD FOR ADAPTIVE PATH PLANNING

Regina Estkowski, et al.
RTN-208PUS

21/21

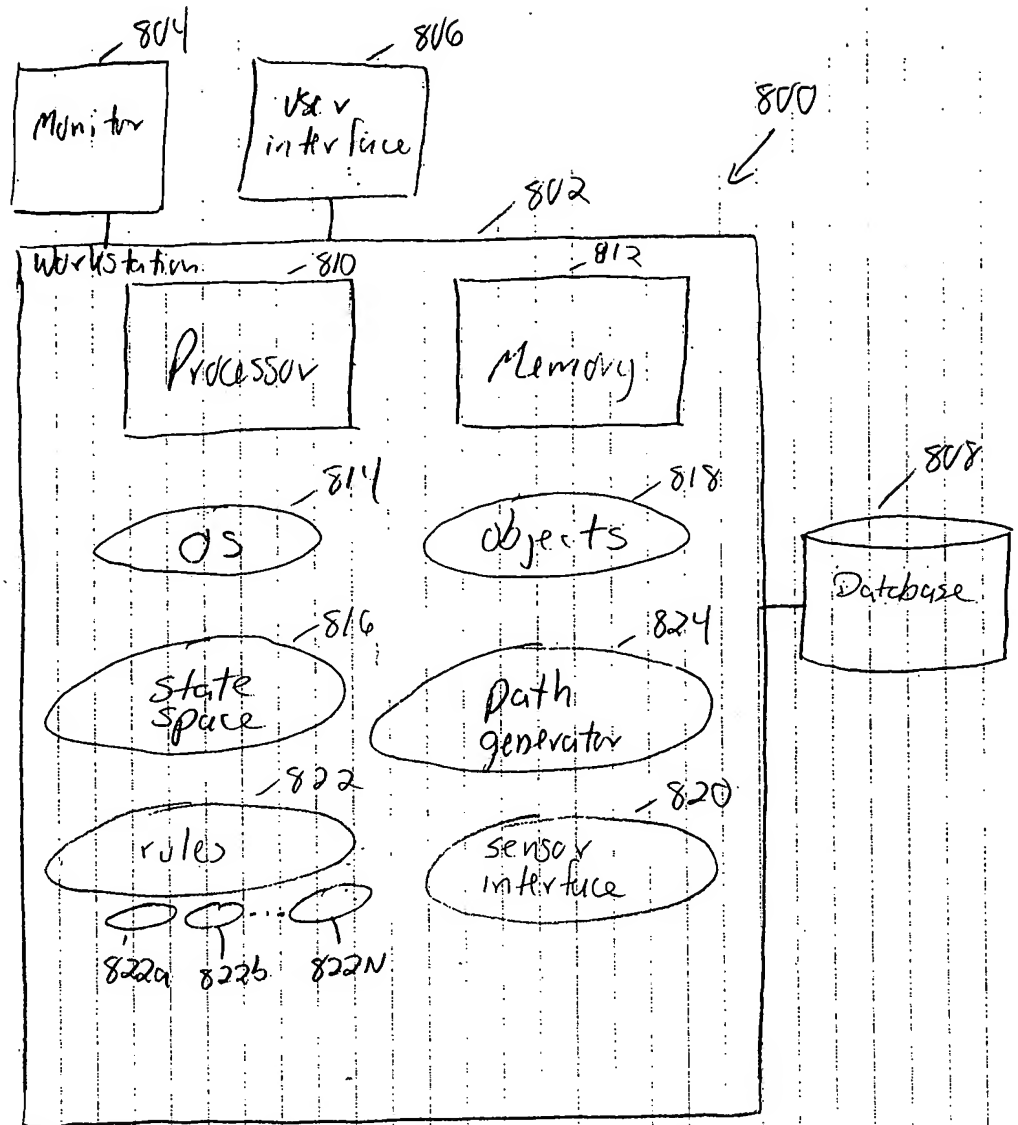


FIG 9